Getting Started with JClass DesktopViews

Welcome
Installation ■ *Product Overviews*
JClass and Your IDE ■ *Technical Support*

260 King Street East
Toronto, Ontario, Canada M5A 4L5
(416) 594-1026
www.sitraka.com

March 2002

RefNo: STJCGS60

Table of Contents

Welcome to JClass DesktopViews 6.0

Introduction

Thank you for purchasing or evaluating JClass DesktopViews! The JClass family of Java components rapidly accelerates the development process by helping Java developers build and maintain applications that use Swing/JFC-based graphical user interfaces.

This booklet provides everything you need to get started with JClass DesktopViews – installation, product overviews, where to find sample code and documentation, and adding JClass DesktopViews to Integrated Development Environments (IDEs).

JClass DesktopViews 6.0 Features

JClass DesktopViews 6.0 continues to support new and emerging Java technologies and provides new features and components to support developers creating enterprise-level Java applications. JClass DesktopViews has one common license and a convenient all-in-one install program. The following provides an overview of the significant new features and improvements contained in JClass 6.0.

JDK 1.4 support – all JClass products now support JDK 1.4!
Note that the Java 3D version of JClass Chart 3D is supported only for JDK 1.3.1 with Java 3D 1.2.1_03 on Windows NT with OpenGL with an ATI RAGE XL PCI video card with these drivers: ati2mpad.sys 4.00.1381.1006 and ati2drad.dll 4.0.0.

JClass Chart – Three new charting types have been added: Polar, Radar, and Area Radar

JClass Chart and JClass LiveTable support XML – developers can populate charts and tables with data from an XML-based data source.

JClass LiveTable and PageLayout have mathematical capabilities – developers can add spreadsheet functionality and evaluation of mathematical formulae to tables display and report output.

JClass Elements contains new Gauge components, both circular and linear– unique and versatile JavaBeans that can function as either an interactive switch/dial or a non-interactive display meter.

For complete details on the new features and improvements contained in this release, please see the JClass readme.

JClass DesktopViews 6.0 Installation CD

If you purchased a physical kit, your JClass installation CD contains:

- JClass Chart 6.0
- JClass Chart 3D 6.0
- JClass DataSource 6.0
- JClass Elements 6.0
- JClass Field 6.0
- JClass HiGrid 6.0
- JClass JarMaster 6.0
- JClass LiveTable 6.0
- JClass PageLayout 6.0

The CD contains products packaged for installation on Microsoft Windows, UNIX, and other development platforms. Installation procedures are outlined in "Installing JClass DesktopViews" on page 5.

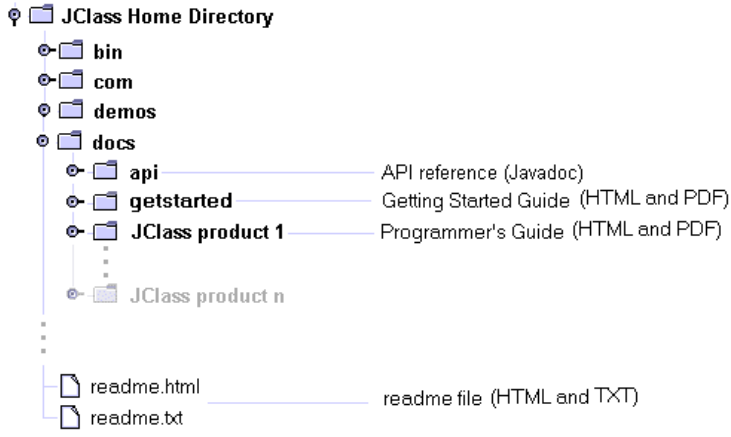
Documentation Resources

JClass DesktopViews comes with a full set of documentation:

- *Getting Started Guide* (HTML and PDF) – this booklet
- *Programmer's Guide* (HTML and PDF) for each JClass DesktopViews product – contains detailed how-to's for programming tasks
- API reference documentation (Javadoc)
- readme (HTML and TXT)

Please note that the *Programmer's Guide* for JClass JarMaster is available only in HTML.

All documentation is installed automatically when you install JClass DesktopViews.



Free JClass Evaluations

It's easy to evaluate any JClass product! You can obtain evaluation versions of any JClass product from the Sitraka Web site at *http://www.sitraka.com*. JClass evaluation products expire after 15 days.

Installing JClass DesktopViews

Startup Checklist

Before installing any JClass product, ensure that your system and development environment are set up correctly. You should be able to create and run simple Java applications before installing JClass products.

Java 2 Development

- JDK 1.2.2 (or higher), **or**
- a Java IDE that supports Java 2

Java Interpreter

- The Java interpreter in your path and the CLASSPATH variable and JAVA_HOME variable must be properly set. (The JAVA_HOME variable must be set to the location of the JDK you are using.) Follow the CLASSPATH setup instructions included with the JDK or IDE that you are using.

License

- You will need a license file. Please see the Licensing section, below.

Licensing

In order to use JClass DesktopViews, you need a valid license file (either evaluation or permanent). This license file (for instance, *jclass-chart.license* or *jclass-desktopviews.license*) is sent to you from Sitraka. Once you receive this file, use the JClass License Wizard to install the license file into the licenses database. The licenses database – called *jclass-desktopviews.licenses* – will be created the first time that you add a license with the License Wizard and then will be found in *JCLASS_HOME/licenses/*.

Note that the JClass License Wizard, which requires a minimum JDK of 1.2, contains online Help to assist you through this installation process.

In order for your JClass DesktopViews product(s) to run, the database license file (*jclass-desktopviews.licenses*) needs to be in your CLASSPATH. **Note that one license file can cover multiple hosts and multiple CPUs**.

When you install JClass DesktopViews product(s), a License Wizard will appear. You must have a valid license for the product(s) you wish to install. If you do not have a valid license, you will not be allowed to install the product. If you wish to add/remove licenses to the license database, you can access the License Wizard by running *LicenseWizard.bat* (Windows installation) or *LicenseWizard.sh* (non-Windows installation), found in *JCLASS_HOME/bin/*, where *JCLASS_HOME* is the directory where you installed the JClass DesktopViews product(s).

Licenses are transferable; please visit the Sitraka Web site at *http://www.sitraka.com/software/support/jclass/tsjclasssupport.html*, email *jclass_support@sitraka.com*, or telephone 1-800-663-4723.

If you try to use a JClass DesktopViews product without a valid license file, the Bean properties in design time mode will not be visible for any IDEs. Thus, you will not see any JClass DesktopViews properties in an IDE's property editor. To purchase any JClass product, please contact Sitraka at *http://www.sitraka.com/buy/* or telephone 1-800-663-4723.

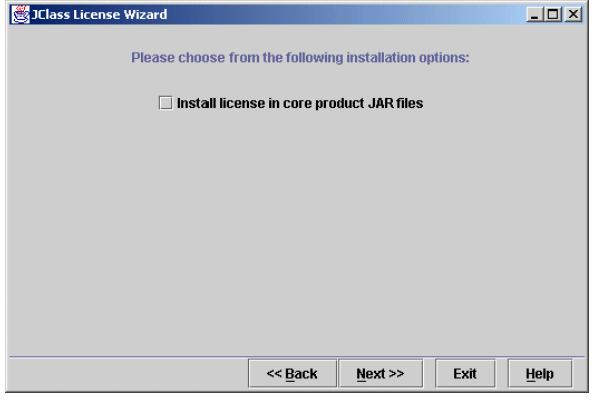
Quick-Start Steps For Adding Your License – Windows and Non-Windows Installation

1. Obtain your license file (for instance, *jclass-chart.license* or *jclass-desktopviews.license*) from Sitraka.
2. During JClass DesktopViews installation, the JClass License Wizard will automatically appear.
3. Use the JClass License Wizard to install the license file into the licenses database. The licenses database will be created the first time that you add a license with the License Wizard and then will be found in *JCLASS_HOME/licenses/*.
4. Add the licenses database file (*jclass-desktopviews.licenses*) somewhere in your CLASSPATH. It is convenient to add your database license file to the JClass DesktopViews product's JAR file (see Adding a License File to JClass JARs); that way, you can simply add that JAR file to your CLASSPATH.

Adding a License File to JClass JARs

Using the JClass License Wizard, you may also install your license file into JClass DesktopViews JAR files.

Once you have selected the product(s) you want to install, the License Wizard allows you to choose whether you want to add your license file to JClass DesktopViews JARs.



If you choose not to add your license file to JClass DesktopViews JARs with the License Wizard during the installation and later decide to add them, simply run the License Wizard again and add them.

JClass License Agreements

JClass License Agreements can be found online at
http://www.sitraka.com/software/support/jclass/tsjclasslicensing.html

Windows Installation

1 **Insert the JClass 6.0 Product CD, or download the installation executable.**
You must have JDK 1.2.2 (or higher) in order to install JClass DesktopViews.

2 **Install the product files.**
If you purchased the product online, you can simply run the executable file you downloaded, for example, *jcdesktopviews600.exe*. Once running, follow the instructions given by the installation program.

If you purchased the CD, you can select and install JClass DesktopViews products with the install program that runs automatically if *autorun* is enabled on your system. If it doesn't run automatically, execute the *autorun.exe* program on the CD. You will see the screen that appears below:

3 **Ensure that the CLASSPATH is set correctly.**
The Windows installation automatically sets the CLASSPATH variable on your system. In Windows 95/98/Me, CLASSPATH is set in the *autoexec.bat* file. In Windows NT, CLASSPATH is set in the *Control Panel* | *System* | *Environment* dialog (*Control Panel* | *System* | *Advanced* | *Environment Variables* dialog on Windows 2000 or Windows XP). You may want to check to ensure that the path is correct.

For detailed instructions on setting CLASSPATH, refer to "Setting the CLASSPATH Environment Variable" on page 10.

4 **Ensure that JCLASS_HOME is set correctly.**
All the JClass DesktopViews demos and examples use the JCLASS_HOME environment variable to find the resources they need to run. JCLASS_HOME should be set correctly by the installation program. If not, you can set it manually at a command prompt. Example of manual installation:

```
set JCLASS_HOME = C:\JClassDesktopViews
```

where `C:\JClassDesktopViews` is the JClass installation directory.

Non-Windows Installation

1 **Insert the JClass 6.0 Product CD, or download the installation class.**
You must have JDK 1.2.2 (or higher) in order to install JClass DesktopViews

2 **Install the product files.**
Install JClass products for which you have a license by running *jcdesktopviews600.jar*

Once in the product directory, run the install class, with the `java` command (JDK 1.2.2 or higher required):

```
java -jar jcdesktopviews600.jar
```
or
```
java -cp jcdesktopviews600.jar run
```

The installation program will guide you through the rest of the installation.

3 **Point the CLASSPATH to installed classes.**
Once the JClass product is installed on the system, you will have to set the CLASSPATH environment variable to point to the product's JAR (eg, *jcchart.jar*). For more information on setting the CLASSPATH, refer to "Setting the CLASSPATH Environment Variable" on page 10.

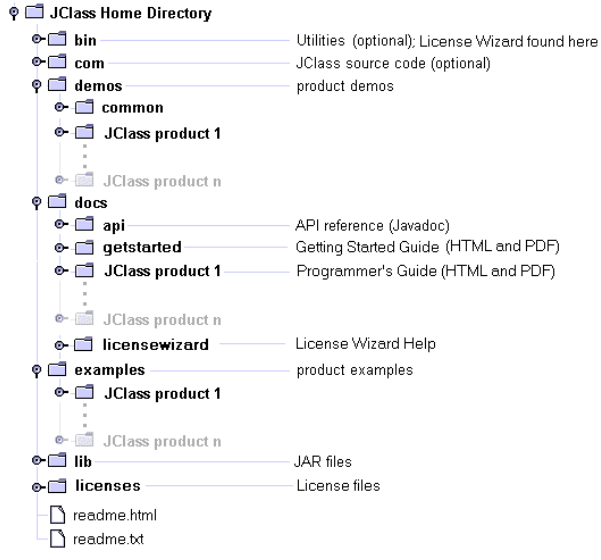
4 **Ensure that JCLASS_HOME is set correctly.**
Many JClass demos and examples use JCLASS_HOME to find the resources they need to run. JCLASS_HOME should be set correctly by the installation program. If not, you can set it manually at a command prompt, for example:

```
setenv JCLASS_HOME /JClassDesktopViews
```

where `/JClassDesktopViews` is the JClass installation directory.

Installed Directories

The following directory structure is created on your system when you install JClass DesktopViews.



Setting the CLASSPATH Environment Variable

The Java Virtual Machine (JVM) and other applications use the CLASSPATH environment variable to locate user-defined classes. You should ensure that the CLASSPATH points to the location of the JClass DesktopViews JAR files (and any other classes you develop). The Windows installation program does this automatically. UNIX users must manually configure their CLASSPATH.

Two entries should be part of the CLASSPATH – one specifying the JClass product classes (a JAR file located in the product's *lib*\ directory), and one specifying the installation directory (necessary to run JClass example and demo programs). You should not need to extract the JAR archive to develop with JClass DesktopViews products.

For example, if you installed JClass Chart on a Windows machine in *C:\JCLASS_HOME*, the CLASSPATH would include the following:

```
C:\JCLASS_HOME\lib\jcchart.jar;C:\JCLASS_HOME\
```

Note: In code snippets using a JAR file we show *jcchart.jar*, but you need to select the version (for example, *jcchartjb.jar*) that best suits your needs. Please see Selecting the Correct JAR File for full details.

To determine the current CLASSPATH, enter the following at a command prompt:

Windows – `echo %CLASSPATH%`
UNIX – `echo $CLASSPATH`

Some CLASSPATH specification tips:

- Each entry is separated by a semicolon (Windows) or a colon (UNIX).
- An entry is typically a root directory to search through for *.class* files (if a class is part of a package, each level in the package is treated as a subdirectory from here), for example, `C:\JCLASS_HOME`.
- Entries can also specify a JAR file containing archived classes, for example, `C:\JCLASS_HOME\lib\jcchart.jar`.
- Add a period (`.`) to the CLASSPATH to include the current directory.
- Setting the CLASSPATH in a startup file causes it to be used when running Web browsers and other applications for your entire session.

Setting the CLASSPATH in Windows

The Windows-based setup program automatically adds your JClass DesktopViews product information to the CLASSPATH during installation. The following instructions are provided in case you need to configure the CLASSPATH manually.

Windows 95, Windows 98, and Windows Millennium

To include JClass DesktopViews products in your CLASSPATH, your *autoexec.bat* file must include the location of your JClass JARs and the home directory. For example, to include JClass Chart in the CLASSPATH:

```
set CLASSPATH=%CLASSPATH%;C:\JCLASS_HOME\lib\jcchart.jar;
 C:\JCLASS_HOME;
```

Restart Windows to make the change take effect.

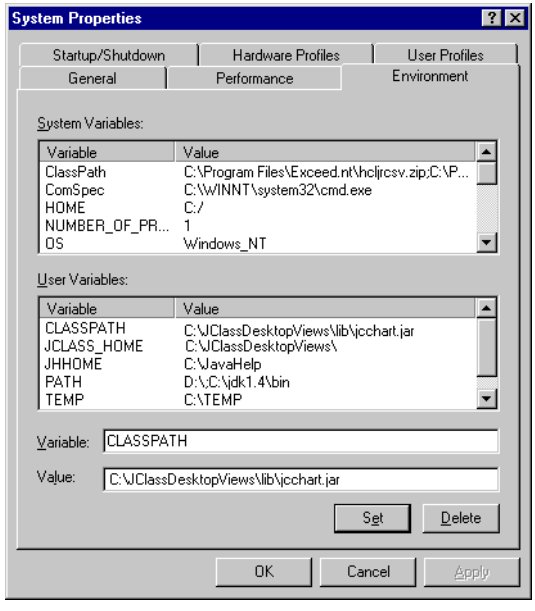
Windows NT, Windows 2000, and Windows XP

Open the **Control Panel** and select **System**. Locate the environment variables on the **Environment** tab (NT) or **Advanced** tab (Windows 2000 and Windows XP).

Look for a CLASSPATH environment variable (if it doesn't exist, create it), and make sure that the JClass JAR location and home location are in the variable. For example, to include JClass Chart in the CLASSPATH, add the following value to the variable:

```
[existing-classes]; C:\JCLASS_HOME\lib\jcchart.jar;
    C:\JCLASS_HOME;
```

The following illustrates a CLASSPATH setting on Windows NT; your actual setting may vary or have additional directories/JAR files.



Setting the CLASSPATH in UNIX

In UNIX, you must manually configure the CLASSPATH environment variable before you can start using any JClass DesktopViews product. The CLASSPATH must point to the location of the classes and installation directory.

For example, to set the CLASSPATH for JClass Chart, add a *setenv* command to your startup file (such as *.cshrc*) to set CLASSPATH to point to the classes, for example:

```
setenv CLASSPATH
   /JCLASS_HOME/lib/jcchart.jar:
   JCLASS_HOME
```

If you are using the Bourne Shell, the commands are:

```
CLASSPATH=$CLASSPATH:/JCLASS_HOME/lib/jcchart.jar:
   JCLASS_HOME
export CLASSPATH
```

Selecting the Correct JAR File

Several JClass products have versions that are optimized for databinding with an IDE or with JClass DataSource. The following table helps determine the JClass product JAR file to install for use with your application.

Platform	Standard	Databinding with Borland JBuilder 5	Databinding with JClass DataSource
Java 2 (JDK 1.2.2+)	jcchart.jar jcchart3dj2d.jar jcchart3dj3d.jar jcdatasource.jar jcelements.jar jcfield.jar jchigrid.jar jcpagelayout.jar jctable.jar	jcchartjb.jar jcfieldjb.jar jctablejb.jar	jcchartds.jar jcfieldds.jar jctableds.jar

Running Demos and Examples

JClass products include an Example & Demo Gallery. Demos are meant to show real-world uses of the product, while examples provide short, simple code examples for particular components. When you install a JClass product, the demos and examples of that product will automatically be installed.

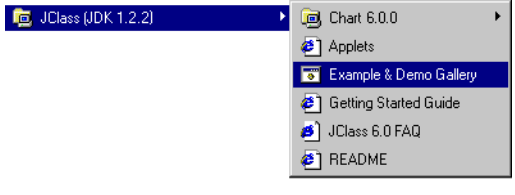
Your CLASSPATH and JCLASS_HOME environment variables must be set correctly in order to run the demos and examples.

As an example, the following instructions show how to run a JClass Chart demo on Windows.

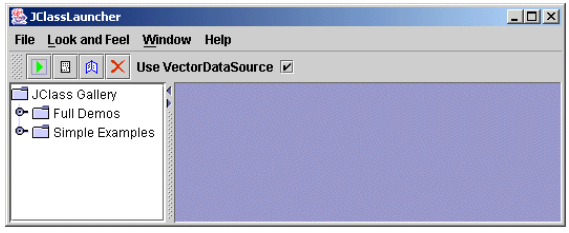
Note: UNIX (as well as Windows) users can run the Example & Demo Gallery via a batch file/shell script. You can run the launchers by entering the following at a command prompt:

```
cd JCLASS_HOME/demos/common
run
```

1 **Choose Start | Programs | JClass Product | Example & Demo Gallery.**

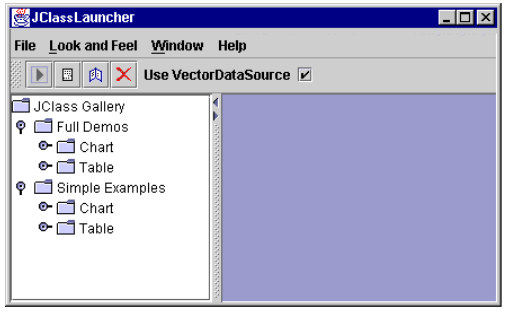


A DOS window will briefly appear, and then the Gallery window will open.



2 Choose the JClass product that you wish to view.

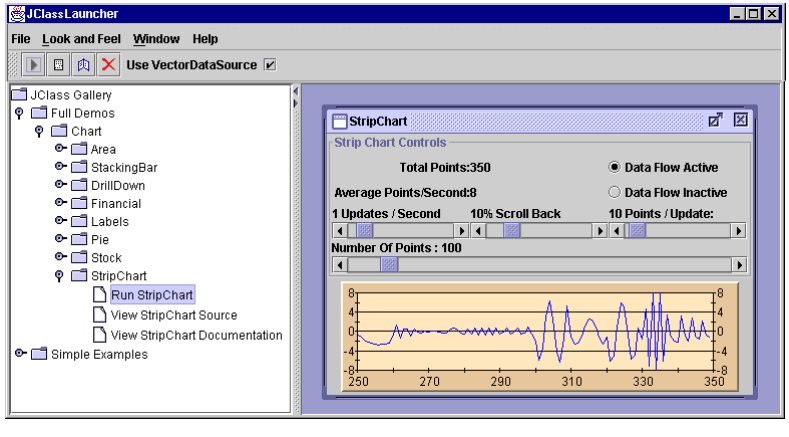
When you open the Full Demos and Simple Examples directories, each JClass product that you installed will have a folder. In this example, the user has installed JClass Chart and JClass LiveTable.



3 Make a selection from the Full Demos or Simple Examples folder.

There are three options below each demo and example – you can run the demo or example, view the original Java source code, or view the associated documentation. You may need to resize the window to fit the demo.

Note that you can run multiple demos or examples simultaneously.

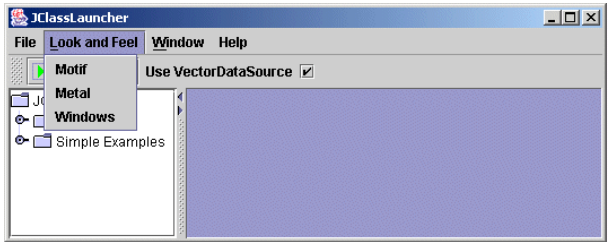


4 Try the example or demo's active features.

As an example, the demo featured in step 3 allows you to enable or disable the data flow, and to change the number of updates per second, the percentage scroll back, the number of points per update and the overall number of points. The chart updates on the fly.

5 Change the Look and Feel, if you wish.

From the Look and Feel menu, select Windows, Motif, or Metal. You can change the Look and Feel on the fly.



Product Overviews

JClass Chart 3D ■ JClass Chart
JClass Elements ■ JClass Field
JClass LiveTable ■ JClass PageLayout
JClass JarMaster ■ JClass HiGrid ■ JClass DataSource

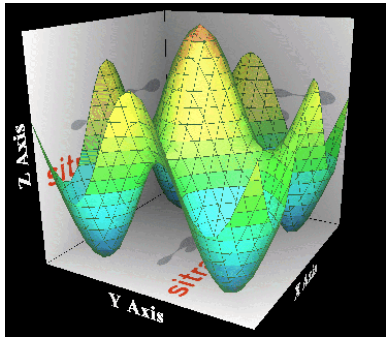
This chapter provides a high-level introduction to each JClass product, including features and quick-start suggestions.

Please remember that JClass DesktopViews comes with a full set of documentation:

- *Programmer's Guide* (HTML and PDF) for each JClass DesktopViews product; this user's guide contains detailed how-to's for programming tasks
- API reference documentation (Javadoc)
- *Getting Started Guide* (HTML and PDF)
- readme (HTML and TXT)

For more details, please see Documentation Resources.

JClass Chart 3D



Three-Dimensional Charting Tools and Beans

With JClass Chart 3D, you can develop sophisticated charting applications with components, or develop fast, effective chart solutions with the JCChart3dJava2d and JCChart3dJava3d JavaBeans.

Professional Capabilities

JClass Chart 3D adds powerful, three-dimensional graphing capabilities to Java. You can use multiple chart types in unlimited ways to present your data.

Business and Scientific Charts

Take your pick of multiple chart configurations with JClass Chart 3D, including:

- Mesh, surface, contour and zone drawing
- Adjustable contour styles and line widths
- 3D rotation and perspective
- Customizable header, footer, and legend styles

Flexible Data Sources

Use the pre-built data sources included with JClass Chart 3D to read in data from memory, a file, a `ResultSet`, and an XML source. And if you want total flexibility, you can even build custom data sources to retrieve data from remote servers or real-time information streams.

Fully Interactive

JClass Chart 3D is ideal for decision support applications because it has built-in support for common chart interactions, like panning, zooming, rotating and picking. It's easy to enable or disable these actions for your users and you have the flexibility to bind to any mouse button and modifier key combination. With JClass Chart 3D's map and pick functions you can enable applications to provide feedback when the user clicks the mouse anywhere on a graph. Transpose, drill down, edit and scale with any chart type to give your users the ultimate in interactive capabilities.

Easily Customizable

There is a customizer similar to the one in JClass Chart. JClass Chart 3D gives you great looking defaults for your charts. Also, it's easy to customize the appearance of your chart axes, headers, footers, legends and data labels. Choose between complete control over text and label positioning or letting JClass Chart's 3D dynamic defaults take over.

Quick-Start Suggestions

The easiest way to familiarize yourself with JClass Chart 3D is to load it into an IDE, then develop a basic charting application.

Using the Chart 3D JavaBean

To use JClass Chart 3D in your IDE, follow your IDE's instructions on loading a Bean component, and ensure that your CLASSPATH is set properly.

Once you have the Beans loaded into your IDE, you can view and change the properties.

Using JClass Chart 3D Programmatically

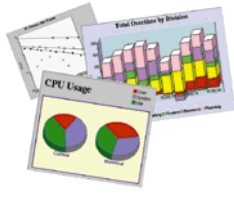
In order to use JClass Chart 3D programmatically, make sure that your `CLASSPATH` points to the JClass Chart 3D installation directory and to any JAR files you might require (*jcchart3dj3d.jar*, *jcchart3dj2d.jar, vecmath.jar*, *jaxp.jar*, *crimson.jar*). You can then create a simple charting application with the main charting class, `JCChart3D`. The *jcchart3dj2d.jar* JAR contains classes that are restricted to the Java 2D API, while the *jcchart3dj3d.jar* JAR contains classes for both the Java 2D and Java 3D APIs.

Browse the Demos and Examples

There are many JClass Chart 3D demos illustrating a wide variety of uses and solutions. They range in complexity and functionality, and they make use of most features of JClass Chart 3D. Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory.

Please refer to Running Demos and Examples for more information.

JClass Chart



Powerful Components and Easy-to-Use Beans

With JClass Chart, you can develop sophisticated charting applications with components, or develop fast, effective chart solutions with the MultiChart and SimpleChart JavaBeans.

Professional Capabilities

JClass Chart adds powerful graphing capabilities to Java. You can use multiple chart types in unlimited ways to present your data. JClass Chart exposes properties for HTML configuration and gives you smart axes for easy coding. And with unique features like the automatic time-axis, you can let your users manipulate data to see it the way they want to.

Business and Scientific Charts

Take your pick of multiple chart types with JClass Chart, including:

- Polar
- Radar
- Area Radar
- Bar
- Stacking Bar
- Pie
- Hi/Lo
- Hi/Lo Open/Close
- Area & Stacking Area
- Scatter Plot
- Plot
- Line
- Candle

Flexible Data Sources

JClass Chart supports a vast array of data sources including files, sockets, real-time feeds and even shares data with JClass LiveTable. With its MVC (Model-View-Controller) architecture, JClass Chart makes it easy to bring data in from any source. Use the pre-built data sources included with JClass Chart to read in data from memory, a file, a URL, an XML source, or any arbitrary `InputStream`. And if you want total flexibility, you can even build custom data sources to retrieve data from remote servers or real-time information streams.

Automatic Data-Binding

You can automatically bind JClass Chart to any database using JClass DataSource (the hierarchical data object that works with any JDBC database) or by using Borland JBuilder 5 or 6.

Fully Interactive

JClass Chart is ideal for decision support applications because it has built-in support for common chart interactions, like panning, zooming, rotating and picking. It's easy to enable or disable these actions for your users and you have the flexibility to bind to any mouse button and modifier key combination. With JClass Chart's map and pick functions you can enable applications to provide feedback when the user clicks the mouse anywhere on a graph. Transpose, drill down, edit and scale with any chart type to give your users the ultimate in interactive capabilities.

Easily Customizable

JClass Chart gives you great looking defaults for your charts. Plus, it's easy to customize the appearance of your chart axes, headers, footers, legends and data labels. Advanced features like reversed logarithmic and multiple Y-axes are also supported and you have lots of formatting options, including scientific notation, percents, and currencies. Choose between complete control over text and label positioning or letting JClass Chart's dynamic defaults take over. The built-in JavaBean customizer allows JClass Chart objects to be manipulated at design time or run-time and lets you modify and save charts using the chart applet HTML format.

Quick-Start Suggestions

The easiest way to familiarize yourself with JClass Chart is to load either the SimpleChart or MultiChart Bean into an IDE, and develop a basic charting application.

There are two tutorials in the *Programmer's Guide*: "SimpleChart Bean Tutorial" and "Chart Programming Tutorial". You can also browse the demo and example directories to see how JClass Chart has been used to create solutions to real-world problems.

Using SimpleChart

To use SimpleChart in your IDE, follow your IDE's instructions on loading a Bean component, and ensure that your CLASSPATH is set properly.

Once you have the Beans loaded into your IDE, you can go through the *SimpleChart Bean Tutorial* chapter in the manual.

Using MultiChart

To explore more advanced charting features, use the `MultiChart` Bean, and refer to the *MultiChart* chapter in the manual.

Using JClass Chart Programmatically

In order to use JClass Chart programmatically, make sure that your CLASSPATH points to the JClass Chart installation directory and to any JAR files you might require (*jcchart.jar*, *jaxp.jar*, *crimson.jar*). You can then create a simple charting application with the main charting class, `JCChart`, by following the tutorial.

Browse the Demos and Examples

There are dozens of JClass Chart demos, illustrating a wide variety of uses and solutions. They range in complexity and functionality, and they make use of most features of JClass Chart. Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory.

For examples of charting applications and data sources, refer to the *JCLASS_HOME\examples* directory.

Please refer to Running Demos and Examples for more information.

JClass Elements

JClass Elements is a collection of pre-built, pre-tested, GUI JavaBean components and utilities designed to enhance and extend Swing.

Powerful GUI Beans

JClass Elements has over 15 components and layout managers for Java GUI development, plus an equivalent number of utilities. JClass Elements allows you to create applications that aren't possible using Swing alone.

Extended Graphical Control

JClass Elements has the classes you need for creating compact multiple document interfaces, friendly wizards, font chooser dialogs, a versatile circular gauge component, a component that's a combination of a tree and a table, and more.

Multi-Document Interface

Easily add the power of MDI with this component that extends Swing's JDesktop view, enabling you to:

- Optimize screen real estate
- Switch to any internal frame or select from tiling algorithms using an automatic localized menu

Powerful Wizards and Managers

A powerful combination of managers and tools that includes:

- Wizard Manager for easy creation of Wizard components
- Align and Grid layout managers for easy form creation
- Springs and braces-based layout for advanced component layout
- Extended Action model that supports mnemonics and accelerators
- Easy-to-use HTML panel that follows links
- Sorting tools, including a shrink-wrapped user-sortable version of Swing's JTable

Robust Value-Added Components and Beans

- Circular and linear gauges that can function as either interactive switches/dials or non-interactive display meters
- Multi-column outliner designed to use an extended Swing tree data model
- Tree explorer, a JTree/JTable combination that uses the same data model as the Multi-column outliner
- Spin boxes for a variety of core Java types
- Multi-select list box for easy multi-item list selection
- Progress helper that displays a cancelable progress bar during any task
- Font choosers for toolbars and panes
- Message helper that displays errors and warnings
- Checkbox list for automatic generation of a group of radio buttons from a list
- Date chooser
- And many more!

JClass Elements gives you all the Swing enhancements you need for real-world application development.

Quick-Start Suggestions

The easiest way to familiarize yourself with the JavaBeans components in JClass Elements is to load one into an IDE and view its properties. You will want to browse the demos, examples and the manual to learn about JClass Elements's utility classes.

Using JClass Elements in an IDE

Many components in Elements are JavaBeans. To use the Beans in an IDE, follow your IDE's instructions on loading components into your environment. Then you can create a simple application by dragging components into a frame or panel. For more information, refer to the next chapter, JClass DesktopViews and Your IDE.

Using JClass Elements Programmatically

In order to program with JClass Elements programmatically, make sure that your CLASSPATH points to your installed JClass Elements classes.

The *Programmer's Guide* is a great starting point for learning JClass Elements's functionality. Each component or utility class has a section that includes procedural information and code samples. An appendix gives property descriptions for the JavaBean components.

Browse the Demos and Examples

The examples and demos illustrate all JClass Elements components. Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory.

For examples of applications built with JClass Elements, refer to the *JCLASS_HOME\examples* directory.

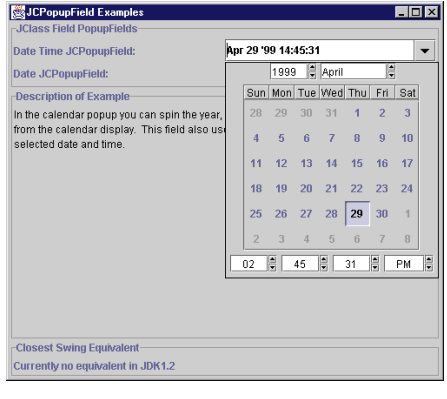
Please refer to Running Demos and Examples for more information.

JClass Field

JClass Field is a set of JavaBeans that allows data input and validation for a wide array of data types. JClass Field makes it easy to create forms and add data input and validation to your applications.

Extensive Control

Includes support for locale-specific date/time and currency for input, formatting and output, float and integers, numeric and string validation with masking. JClass Field also supports picklists, spinboxes, and comboboxes.



Save Time

Using JavaBean technology, JClass Field offers you a choice of interaction techniques not available in other components and is designed to make it easy to add validation fields to your Java programs. JClass Field provides full support for:

- locale-specific date and time
- calendar pop-ups
- pick-lists
- spin-boxes
- combo-boxes with optional automatic completion of text entry

You can use JClass Field to validate:

- strings
- floating point values
- time or date values
- integer values

Automatic Completion of Text in a JCComboField

Allow your users the ease of having the text field in a JCComboField suggest the desired item based on the characters already typed. The field matches what has already been typed against the items in the list model. The user may accept this candidate or type more characters to refine the match.

Edit and Display Masks

JClass Field gives you validation masks in String and Calendar (Date/Time) components to specify which characters are valid at each editable field position.

Validation

JClass Field makes it easy to validate strings entered in a text field, or chosen using spin and combo fields. It supports validation of floating point values in text and spin fields and can validate a time value in text and spin fields.

Calendars

JClass Field includes a general purpose calendar. Use the pop-up calendar control to let your users easily select valid dates. Never waste time writing complex validation code again!

Quick-Start Suggestions

Using JClass Field in an IDE

All JClass Field components are JavaBeans. To use JClass Field in your IDE, make sure that your environment is set up properly, as described in the installation section of this booklet. Then, follow your IDE's instructions on loading a Bean component. For more information, refer to the next chapter, JClass DesktopViews and Your IDE.

Using Components

In order to use the components, make sure that your CLASSPATH points to your installed JClass Field classes. You can then create a simple data entry application with any of the JCField components by modeling it on the example and demo code.

Browse the Demos and Examples

The examples and demos illustrate a wide variety of uses and solutions of JClass Field. They range in complexity and functionality, and they make use of most features. Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory.

For examples of applications built with JClass Field, refer to the *JCLASS_HOME\examples* directory.

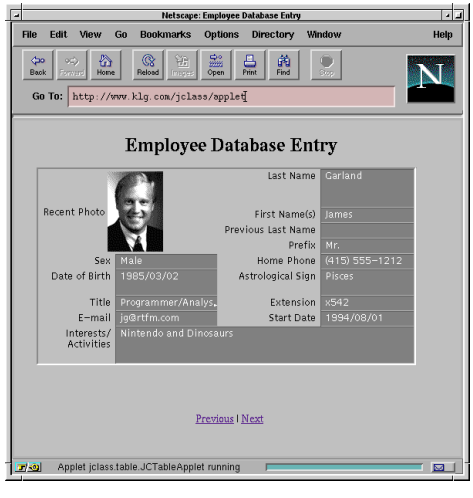
Please refer to Running Demos and Examples for more information.

JClass LiveTable

JClass LiveTable is a high quality grid/table JavaBean for data-driven display.

Highly Configurable

JClass LiveTable provides extensive properties and methods with a wide variety of customization options. It can easily be configured to act as a spreadsheet, data-entry form, multicolumn list or data grid. With JClass LiveTable, the data you display can come from a database, a real-time system, a comma-delimited file, or directly from the Web page that launched the applet. Interactive features such as cell resizing, on-the-fly sorting and searching are easy to use. And since the properties of JClass LiveTable can be changed at run-time you get the ultimate in flexibility and reuse.



Superior, Scalable Performance

JClass LiveTable is optimized to provide superior performance as you scale your operating conditions. It uses a unique queued scrolling system that allows real-time scrolling with smooth repainting. JClass LiveTable supports huge virtual tables so you can choose to deliver only the information the user needs. Fast cell-level redraw gives you maximum efficiency, and with JClass LiveTable you have the ability to turn double-buffering on or off as needed.

Total Control Over Fonts, Color and Images

JClass LiveTable gives you total control over color, font, border styles and shading using a unique style-based property setting mechanism. Your data can mix fonts, color, and images using HTML code.

Any Data from Anywhere

With JClass LiveTable, your tables can be populated with data from a file, a URL via HTTP, an XML source, a relational database via JDBC, supplied programmatically or entered by an end user. Specify cell values as regular text strings or as any arbitrary Java object. It's even easy to use real-time data with a custom data source that notifies the table whenever new data is available.

Automatic Database Binding

JClass LiveTable automatically binds to your database, enabling point-and-click RAD engineering (requires purchase of JClass DataSource, or Borland JBuilder 4 or later).

Customizable Cells

Many of JClass LiveTable's properties can be set on a per-cell basis enabling effortless control over cell content. JClass LiveTable also uses a customizable cell renderer and editor architecture to give you the flexibility to create a table that suits your needs. These can be associated with a cell's data type or with a specific region of the table. Plenty of built-in cell renderers and editors are provided for a number of common data types as well as for specialty functions such as word wrap and viewing booleans as checkboxes. You can specify mathematical formulas in cells that dynamically calculate and display results. You can use JClass Elements, Swing or JClass Field components in cells. JClass Field objects can be used as editors or renderers so you can easily drop in a calendar pop-up or use a spin-box, with minimum overhead and no extra work!

Printing Support

JClass LiveTable fully supports the JDK printing capabilities and extends it to give you print preview functionality, allowing users to view how the table will be printed and paginated before it is sent to a printer. Large tables are automatically spanned across multiple pages and customizable headers and footers are also supported.

Quick-Start Suggestions

The easiest way to familiarize yourself with JClass LiveTable is by loading LiveTable into an IDE and following the LiveTable Bean Tutorial.

There are two tutorials in the *Programmer's Guide*: a Bean tutorial and a component tutorial. You can also browse the demo and example directories, or run their respective launchers, to see how JClass LiveTable has been used to create solutions to real-world problems.

Using Beans

To use JClass LiveTable in your IDE, make sure that your environment is set up properly, as described in the installation section of this booklet. Then, follow your IDE's instructions on loading a Bean component into your IDE. For more information, refer to the next chapter, "JClass DesktopViews and Your IDE".

Once your components are loaded into your IDE, go through the Bean tutorial in "JClass LiveTable and IDEs" in the *Programmer's Guide*.

Using Components

In order to use the components, make sure that your CLASSPATH points to your installed JClass LiveTable directory and to any JAR files you may require (*jctable.jar*, *jaxp.jar*, *crimson.jar*). Then create a simple table application by following the component tutorial in the "Hello Table – A Simple JClass LiveTable Program" chapter in the *Programmer's Guide*.

Browse the Demos and Examples

The JClass LiveTable demos illustrate a wide variety of uses and solutions. They range in complexity and functionality, and they make use of most features. Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory.

For examples of JClass LiveTable applications, refer to the *JCLASS_HOME\examples* directory.

Please refer to Running Demos and Examples for more information.

JClass PageLayout

JClass PageLayout is the ultimate tool for putting your Java application data on paper or on-screen. JClass PageLayout gives developers professional printing power in a set of Java libraries with methods and procedures for adding paginated, formatted, flowed-text, and image output to Java applications.

The Power of Page Templates

Start from any of JClass PageLayout's predefined templates or build your own from scratch. You'll never have to duplicate your efforts, as JClass PageLayout reuses the page templates you design, and applies them to the pages you specify.

Advanced Formatting

Use JClass PageLayout to develop complex layouts quickly and easily that incorporate columns, frames, multiple fonts and styles, automatic page numbering, headers and footers, and sophisticated tables. Versatile text formatting is made easy with font maps, text alignment, paragraph and line spacing, indenting and tab controls. Add color to pages, frames and text, and diagrams with convenient line-drawing and shape library. JClass PageLayout also lets you create sophisticated tables using built-in support for JTable and JDBC tables.

Total Printing Control

With JClass PageLayout, you can convert your JTable, JDBC or JClass LiveTable data directly into printable form. A wide range of output options are available, including Java Printer, Acrobat PDF, PostScript Level 2, or PCL 5. You can also integrate the Swing-based Previewer into your application for previewing formatted output before printing.

The JClass PageLayout Advantage

- Sophisticated page templates
- Landscape or portrait orientations
- Running headers and footers
- Automatic page numbering
- Multiple columns
- Font maps
- Text alignment
- Paragraph spacing
- Line spacing
- Indents

- Tab stops
- Add color to pages, frames, and text
- Line-draw and shape library
- Supports JClass LiveTable, JTable, and JDBC tables
- Powerful mathematical operators for extending and summarizing tabular data
- Import EPS, GIF, JPG, and PCL images
- Print previewing
- Print to any system printer
- PostScript (PS, EPS), Adobe Acrobat (PDF), or HP PCL 5 output

Quick-Start Suggestions

The JClass PageLayout *Programmer's Guide* provides the complete code of the simplest PageLayout program – the *Hello, World* example. The example strips a JClass PageLayout program down to its basics: the template, the printer, the document, and the flow.

Browse the Demos and Examples

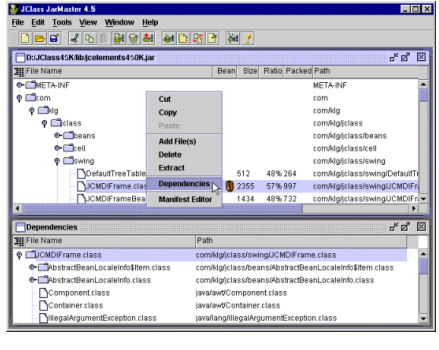
The JClass PageLayout demos illustrate a wide variety of uses and solutions. They range in complexity and functionality, and they make use of most features. Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory.

For simple example code for many applications of JClass PageLayout, refer to the *JCLASS_HOME\examples* directory.

Please refer to Running Demos and Examples for more information.

JClass JarMaster

A powerful Java archive utility, JClass JarMaster dramatically reduces the time required to load and run your finished Java application, because it selects only those Java classes that you need to create a single compact JAR or ZIP file. JClass JarMaster makes it easy to package and manage your classes to ensure fast download.



Rapid JAR Building

Building JARs is a breeze with JClass JarMaster. Using either the GUI or the command line, you can create a single compact JAR or ZIP file quickly and easily.

Advanced Hierarchical Interface

JClass JarMaster's advanced hierarchical interface gives you complete flexibility to add or delete classes, transfer between archives, or refresh the contents of an existing archive based on the contents of your CLASSPATH. Easily drill down to determine dependencies before you create the JAR, or conversely, drill up to determine which classes are dependent on a specified class. You can even edit or explore multiple JARs simultaneously.

Customize Your JAR's Manifest

The JarMaster Bean manifest editor allows you to customize your JAR's manifest, adding or removing Beans as you see fit. You can even modify version information or file dependencies!

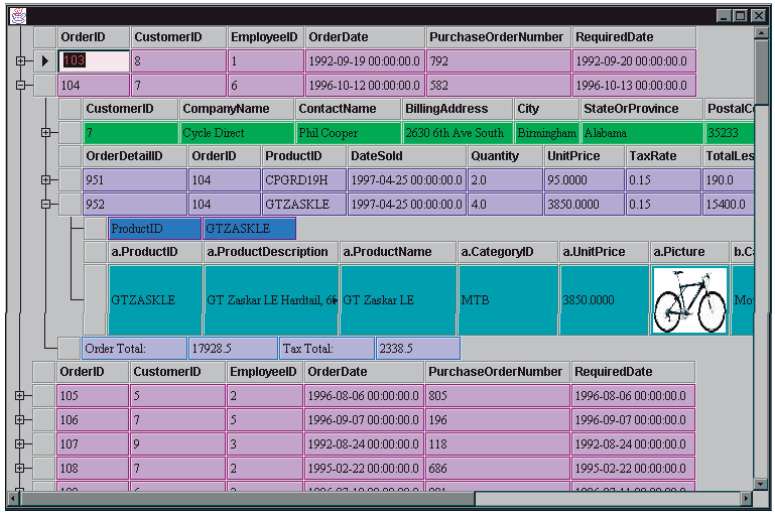
Easy-To-Use Wizard

JClass JarMaster's easy-to-use wizard walks you through the process of building a JAR for your application or library.

JClass HiGrid

Unique RAD Outline-Grid

JClass HiGrid is a unique RAD outline-grid with a hierarchical data source for managing, displaying and updating master-detail relation data in one easy-to-use component. Simply specify master and detail relationships for each level of the tree-view hierarchy, and HiGrid takes care of the rest!



Hierarchical Interface

Unlike most data-bound grids, which can only display records from one table at a time, JClass HiGrid lets you present your users with a multi-level, hierarchically organized dynamic grid display. Many users want to explore hierarchical data relationships that cannot easily be developed using traditional data-binding solutions. They want to drill down to view and edit the detailed data behind the top-level view, but setting up this kind of relationship typically involves writing a significant amount of code. JClass HiGrid lets you do this without writing any code!

Automatic Data Binding

Creating a hierarchical data-bound GUI is as simple as specifying your queries and joins! Choose the detail to be displayed at any given level, and JClass HiGrid's advanced multi-level display makes more information visible to the user. JClass HiGrid includes JClass DataSource, which is capable of binding to any JDBC object and can display dynamic hierarchical data structures in array or unbound mode.

Completely Configurable

JClass HiGrid's built-in customizer lets you configure a grid's appearance by changing its fonts, border, styles and colors. With powerful pluggable cell editors and renderers you get full control over input and display.

Advanced Data Manipulation

JClass HiGrid manages master-detail transactions automatically, or with your own transaction management. Automatically compute summary information to display in footers and use the built-in column sorting and user pop-up menus for more advanced data manipulation.

Quick-Start Suggestions

Browse the Demos and Examples

Run the demos and examples to get a feel for how JClass HiGrid functions. Installed with HiGrid is a demo database that must be set up before the examples will run. For instructions, refer to Installing the Database.

Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory. For examples of JClass HiGrid applications, refer to the *JCLASS_HOME\examples* directory. Please refer to Running Demos and Examples for more information.

Using JClass HiGrid

To use HiGrid in your IDE, or BeanBox, drop `JCHiGrid` into the design area, and click the **GridProperties** property, which brings up a comprehensive property editor used for setting data binding and other properties of the grid.

Binding Other Components to the Data Source

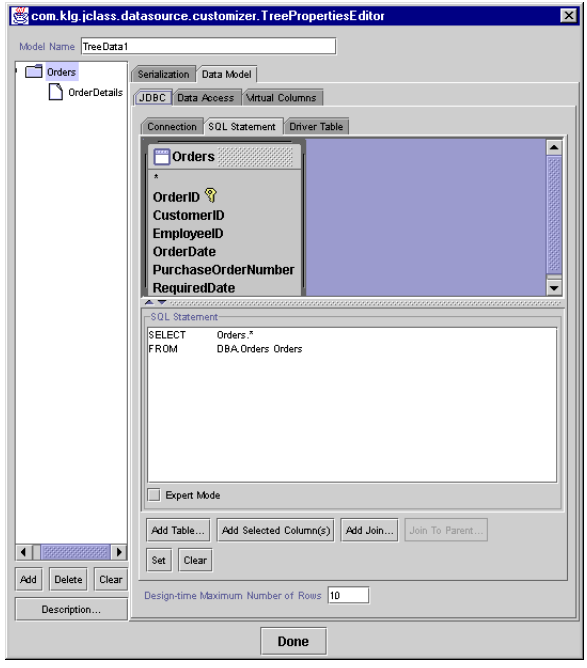
Once data binding has been established, binding lists and other components to the data becomes trivial. Simply drop the component in the design area, and choose which level of data you want to display.

JClass DataSource

With JClass DataSource, you can develop sophisticated applications that connect to databases and access hierarchical data.

A Powerful Tool For Data Access

JClass DataSource gives you a robust hierarchical multiple-platform data source, written in 100% Pure Java. Experience the power of accessing multiple tables in multiple databases by expressing a hierarchy of queries. JClass DataSource represents this hierarchy as a tree so your components can easily navigate the meta-data and bind to a particular table and column.



A JClass DataSource instance can be specified at design time using a rich collection of customizers. The combination of hierarchical multi-platform data access together with JClass components provides maximum flexibility while enhancing ease of use for both API and IDE users. You'll find it easy to represent multiple levels in the database and tie them together in a logical manner. Common drill-down data access becomes trivial!

JClass DataSource provides a common abstraction to multiple database connection techniques. The actual data source can be a JDBC URL, or a JBuilder DataSet.

JClass DataSource includes data-bound versions of popular Swing components such as TextField, ListBox and Label.

Works with Popular JClass Components

JClass DataSource can be used to bind JClass Chart, JClass LiveTable, and JClass Field to databases.

Quick-Start Suggestions

JClass DataSource is distributed with a collection of programs. A demo database is required to work with these examples. Use the following procedure to install and configure your system to use the demo database.

Installing the Database

JClass DataSource's examples use a Microsoft Access(TM) demonstration database called *Demo.mdb*. It resides in the *demos\common\databases* directory along with the sample programs. Thus, the database is installed at the same time as all the other files in JClass DataSource. You can verify its existence by listing the contents of the *demos\common\databases* directory. You can tour the seven tables in this database now if you happen to have Microsoft Access installed on your computer, although this is not a requirement because you will be viewing it using JClass DataSource as soon as you run the example programs.

Setting Up the Data Source

Since any use of this product depends on access to an underlying source of data which is most often going to be a commercial database, you have probably configured your data source already. If not, this section is included here simply because it is one of the first things that must be done if your connection is to be accomplished through a JDBC-ODBC bridge.

It is assumed that you have installed the database product itself prior to beginning the ODBC registration process. In fact, a sample database accompanies JClass DataSource. It is automatically installed along with the class libraries from your distribution medium. Many database products contain similar tutorial databases which you may want to register as well, following the procedure outlined below.

Microsoft Access Driver

To set up a data source called "JClassDemo" in Windows:

1. **Click Start > Settings > Control Panel** to open the *Control Panel* window.
2. Launch the *ODBC Data Source Administrator* dialog by double-clicking the **ODBC** (Windows NT) or **32 Bit ODBC** (Windows 95/98/ME) icon. On Windows 2000 or Windows XP, double-click **Administrative Tools**, then **Data Sources** (**ODBC**).
3. Click **Add** and choose *Microsoft Access Drivers*.
4. Type in the data source name, *JClassDemo*.
5. In the *Database* group of buttons, choose **Select...**
6. Use the Select Database window to choose the location of *demo.mdb*, either by navigating or by typing in the full path and database name.
7. In the *Setup* window, click **Advanced...** to see the *Set Advanced Options* dialog. The Default Authorization group contains a *Login name* field and a *Password* field. The JClassDemo data source uses the default settings, which are **Admin** for the login name and no password.
8. Close all setup windows. Your ODBC data source is configured.

Sybase SQL Anywhere Driver

To set up a data source called "JClassDemoSQLAnywhere" in Windows:

1. Ensure that the Sybase SQL Anywhere driver has been installed.
2. **Click Start > Settings > Control Panel** to open the *Control Panel* window.
3. Launch the *ODBC Data Source Administrator* dialog by double-clicking the **ODBC** (Windows NT) or **32 Bit ODBC** (Windows 95/98/ME) icon. On Windows 2000 or Windows XP, double-click **Administrative Tools**, then **Data Sources (ODBC)**.
4. Click **Add** and choose *Sybase SQL Anywhere 5.x* from the *Create New Data Source* dialog, then click **Finish...**.
5. The *SQL Anywhere ODBC Configuration* window opens. In the *Database Startup* group, browse to the location of the `demo.wrt` database and choose it as the *Database File*.
6. Replace the Data Source Name with *JClassDemoSQLAnywhere*.
7. In the *Connection Information* group of text fields, type **dba** for the *User ID*, and **sql** for the *Password*.
8. Click **OK** and close all *Control Panel* windows.

Setting up the data source for other platforms follows a similar procedure. Consult the platform's documentation for details.

You may need to install one or more driver products to connect your Java programs to various databases. You register these with the *ODBC Data Source Administrator* (or the analogous data source administrator for your platform) following the procedure outlined above. Your application will use the server function of the driver to send SQL statements to the database.

There are many client-server architectures in common use. The procedure given here has assumed a development environment where the database, its drivers, and your code are all on the same machine. If this is not the case, and you are testing your work by connecting to a networked database through a middleware server, you may need the help of your systems administrator to ensure the proper permissions are set and that compatible connection software is installed on all the cooperating machines.

You use the JClass DataSource API to enable your programs to access databases. There is little need for referring to the JDBC API because any reference to this API is made internally. Thus, you use the methods defined in the JClass DataSource API to access the database, perform whatever operations are needed, and display the results.

Browse the Demos and Examples

The examples and demos illustrate JClass DataSource components. Demos were installed with your product, and can be found in the *JCLASS_HOME\demos* directory.

For examples of applications built with JClass DataSource, refer to the *JCLASS_HOME\examples* directory.

Please refer to Running Demos and Examples for more information.

Using JClass DataSource

Once you have run an example to see how JClass DataSource is used, you can drop `JCTreeData` or `JCData` into the design area of your IDE (or BeanBox) and make a connection to your own database.

Your connection won't be very interesting, however, until you can display the information. To do this, you can use JClass Field, JClass Chart, JClass LiveTable, or the components that come packaged with JClass DataSource, such as DSdbNavigator.

JClass DesktopViews and Your IDE

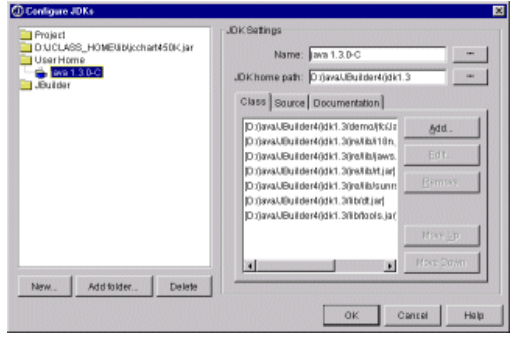
JClass JavaBeans work well with any JavaBeans-compliant Integrated Development Environment (IDE). This chapter provides instructions for adding JClass JavaBeans to Borland JBuilder 5 or higher and IBM VisualAge for Java 3.5 and higher. If you are using a different IDE, refer to its documentation for instructions on adding JavaBeans.

Once the JClass JavaBeans are added to the development environment's component palette, you can use them the same way you use standard AWT or Swing components – adding them to forms, specifying event-handling, and so on.

All environments provide a way to add components contained in a JAR file to their component palette. The exact steps are unique to each environment so the best source for details is the documentation that came with your development environment. The JClass JAR files are located in the *\lib* subdirectory of your JClass installation.

Adding JClass JavaBeans to Borland JBuilder 5 or higher

- We strongly recommend installing JClass DesktopViews products *after* installing JBuilder; this way, each JClass product is added to the Component Palette automatically.
- The following screenshots feature Borland JBuilder version 5. Borland JBuilder 6 may have slightly different menus.



Installing Borland JBuilder, then the JClass product (preferred method)

If you install JClass DesktopViews after you install Borland JBuilder, then the JClass JavaBeans will be automatically integrated in the IDE. This means that the product's JavaBeans will appear on the Component Palette, ready for use.

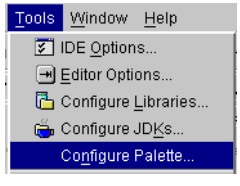
The JClass JavaBeans will be automatically integrated only if you install the executable file (that is, the *.exe* file). If you use a non-Windows installation (that is, you ran the *.class* file), you will need to add the JClass product's JAR file to JBuilder's Component Palette (instructions are outlined in the next section).

Note: Only the 2D version of the JClass Chart 3D JavaBean works in JBuilder, and you will need to add *vecmath.jar* to your project.

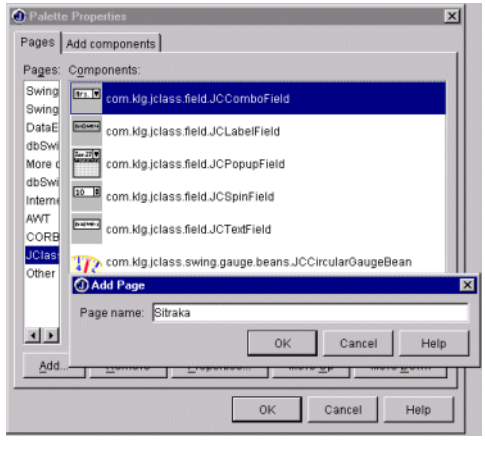
Installing JClass JavaBeans before installing Borland JBuilder

If you install a JClass product before installing JBuilder, follow these directions to add JClass JavaBeans to JBuilder.

1. Make a library available to JBuilder by clicking **Configure Libraries** in the **Tools** menu and navigating to the library you want to add. Alternatively, add the required libraries in the Project Wizard.
2. From JBuilder's **Tools** menu, choose **Configure Palette...** (or right-click the component palette and choose **Properties...**).

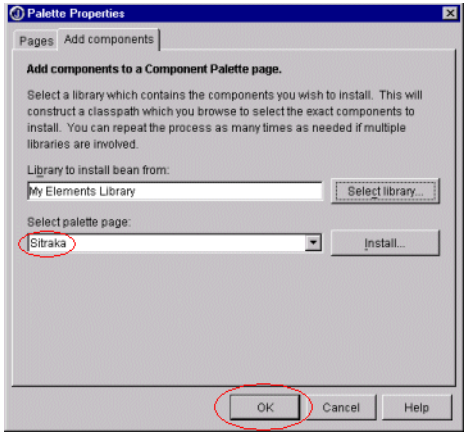


3. From the **Pages** tab of the **Palette Properties** dialog, click **Add...** to create a new page. Label this page, for example, "Sitraka" (alternatively, create a new page for each JClass product that you install).

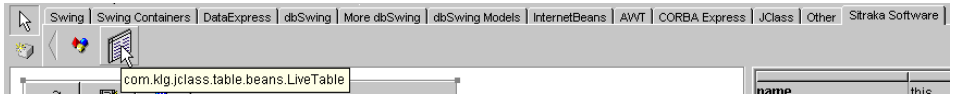


4. Click on this new page name and select the **Add Components** tab of the **Palette Properties** dialog.
5. Click **Select Library** and browse to where you installed your JClass library, then click **Install**. Select the JClass JavaBean class you want to add, or use control-click for multiple selects.
6. Select the component(s) that you wish to install.
7. In the **Select Palette Page** box, select the page where you want this component to be stored (for example, "Sitraka"); press **OK.**

For more information, see the JBuilder help.



You should now see your page name in the component palette. When you click on it, icons for the Beans you added will appear. When you are in JBuilder's Design mode you will be able to drop the components into the frame.



- Please note that limited versions of JClass products come with JBuilder. If you add the full or evaluation version of JClass Chart and JClass LiveTable to JBuilder, the limited versions will remain as separate tabs on the palette. If you would like to remove the limited versions, simply right-click the component palette and choose **Properties...**; highlight the components that you want to delete and click **Remove**, then **OK**. Also, remove the Sitraka libraries.

Upgrading to a Newer Version of a JClass Product

When you install a newer version of a JClass DesktopViews product, the Component Palette is automatically updated to use the new version. However, existing JClass product projects may need to be reconfigured to use the new version as follows (for instance if you installed into a different directory):

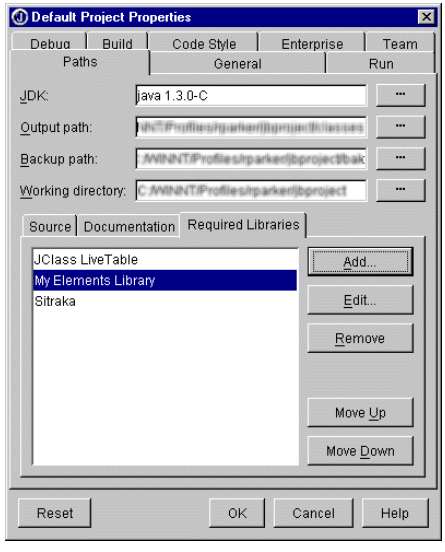
1. With your project open, display the Project Properties dialog (**Project | Default Project Properties...**).
2. Edit the Java libraries list on the **Required Libraries** tab to use the new version of the JClass library.
3. Save your project files.
4. If necessary, edit the project-specific Java Libraries list (**Project | Project Properties...**) to use the new version of the JClass product JAR file for new projects.

See the JBuilder help for complete details. There should generally be no problem using a newer version of a JClass product with an existing application. However, if you do experience problems, you can revert back to the previous version in the Project Properties dialog.

Removing a JClass Product from Borland JBuilder

Using the **Add/Remove Programs** dialog in the **Control Panel** does not remove a JClass product from JBuilder. You must manually configure JBuilder to remove all references to the JClass product:

- Remove the JClass product from the default Java libraries list using the Default Project Properties dialog (**Project | Default Properties...**).

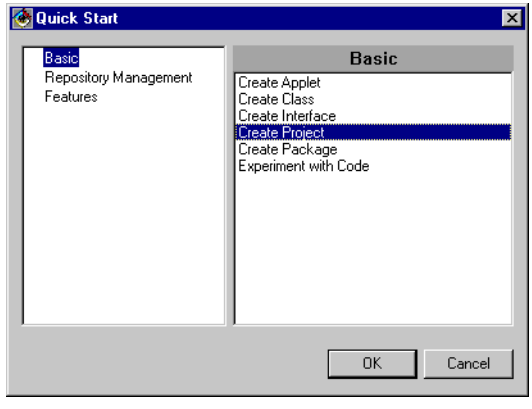


- To remove the JClass product tab from the Palette, right-click the palette, select **Properties...** then click the **Remove** button and press **OK**.

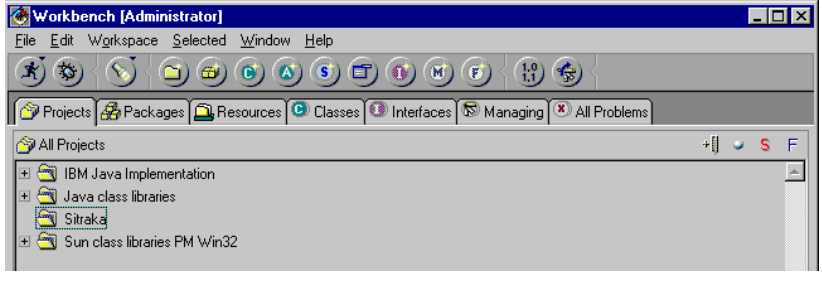
Adding JClass JavaBeans to IBM VisualAge for Java 3.5 and higher

For full details on using JavaBeans with IBM VisualAge for Java 3.5 and higher, please refer to the documentation that came with VisualAge. Essentially, after installing the JavaBeans, create a project and add the JClass JavaBeans to it. Again, for complete details, please refer to the VisualAge documentation. We provide an overview here.

1. From the **File** menu, choose **Quick Start**. In the dialog that pops up, select **Basic** in the left pane, then **Create Project** in the right pane. Click **OK**.

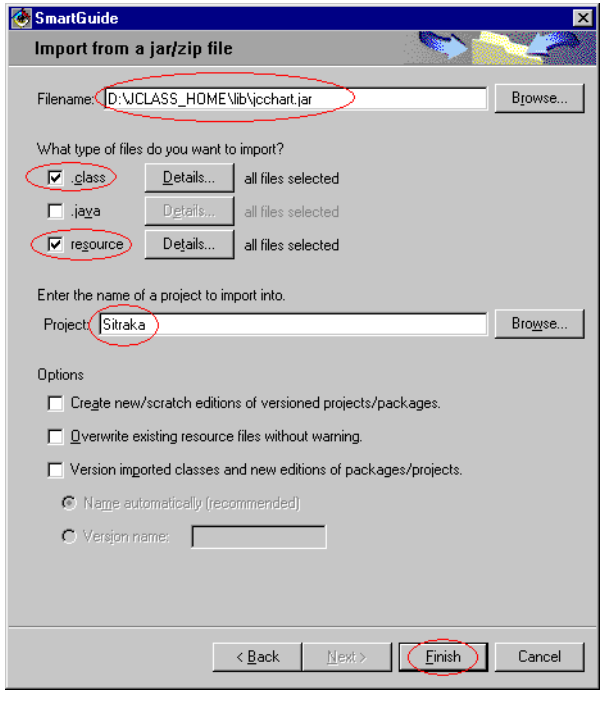


2. The "Add Project" window will automatically open. Type in the name of the project, such as "Sitraka", then click **Finish**. "Sitraka," the project title, will appear on the list of projects displayed when the **Projects** tab of the Workbench is selected.

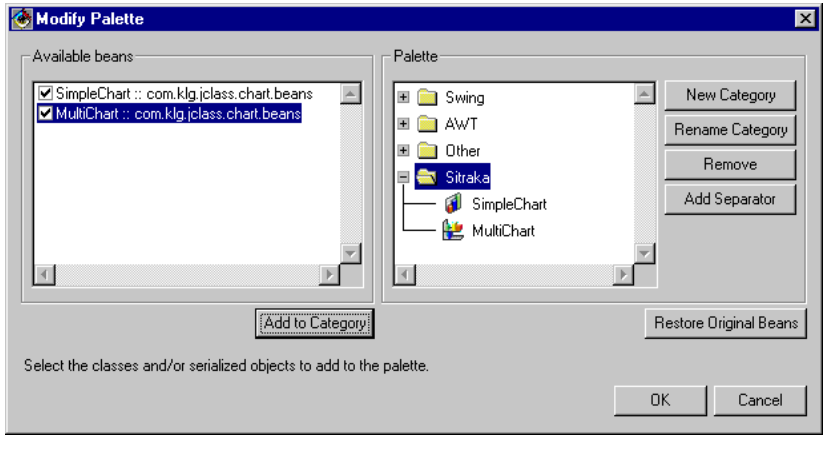


3. To add the JClass product(s)' JavaBeans to the Sitraka project, first click on the Sitraka project in order to select it.
4. From VisualAge's **File** menu, select **Import...**. When the Import window opens, click on the radio button beside **Jar file**, then click **Next**.
5. The "Import from a jar/zip file" window will open. Browse to the *lib*\ directory of your JClass product(s) to select the JAR file you wish to install. Select *jcproduct.jar*, where *product* is the name of the JClass JavaBean package you want to add.

6. Click the box next to ".class" and the box next to "resource", then browse to "Sitraka" as the name of the project into which to import the JAR files. Then click **Finish**.

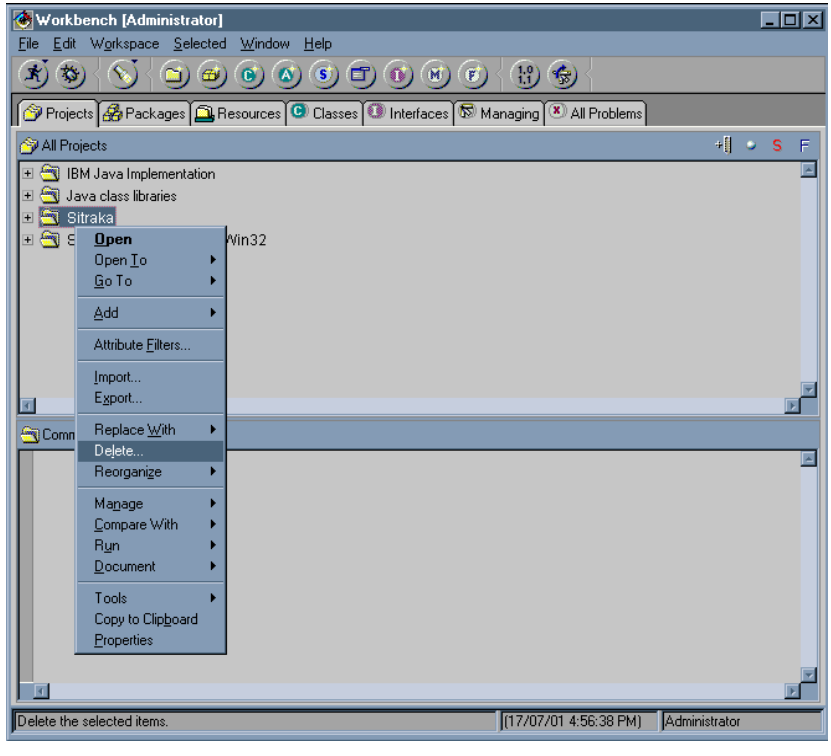


7. The "Modify Palette" window will now appear. Click on the **New Category** button; label it "Sitraka". Next, click the boxes beside the available beans (left pane), then on the newly created Sitraka category (right pane). Click the **Add to Category** button beneath the left pane. Click **OK**. The JClass JavaBeans have now been added and are ready for your use.



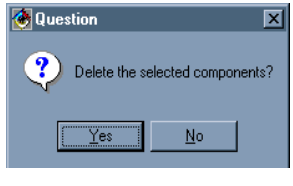
Removing JClass JavaBeans from IBM VisualAge

1. To remove JClass Beans from VisualAge, right click on the Bean you would like to remove in the Workbench, and select Delete from the drop-down menu.



Note: You can choose to remove either a Bean, a package, or a project with this method.

2. When the **Question** window appears, click **Yes**.

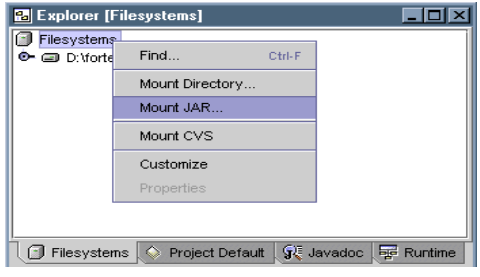


Adding JClass JavaBeans to Sun's Forte for Java 3.0

For full details on using JavaBeans with Sun's Forte for Java 3.0, please refer to the documentation provided with Forte. We provide an overview here.

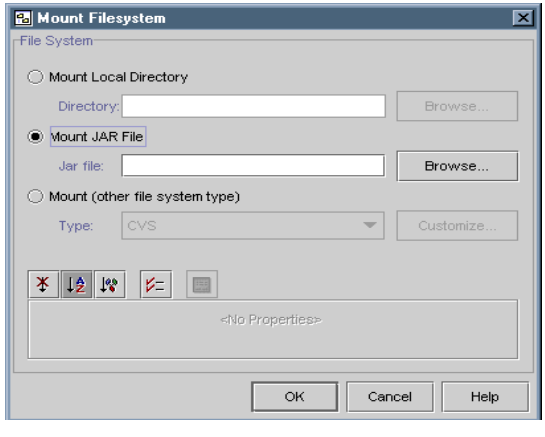
1. The first step is to mount the JAR from which you will take the JavaBeans. This can be done in one of two ways:

   - In Forte's **Explorer** window, right click on **Filesystem** and select **Mount JAR...** from the list. Navigate to the location of the JClass JAR files and select the one(s) you would like to install. Click **Mount**.



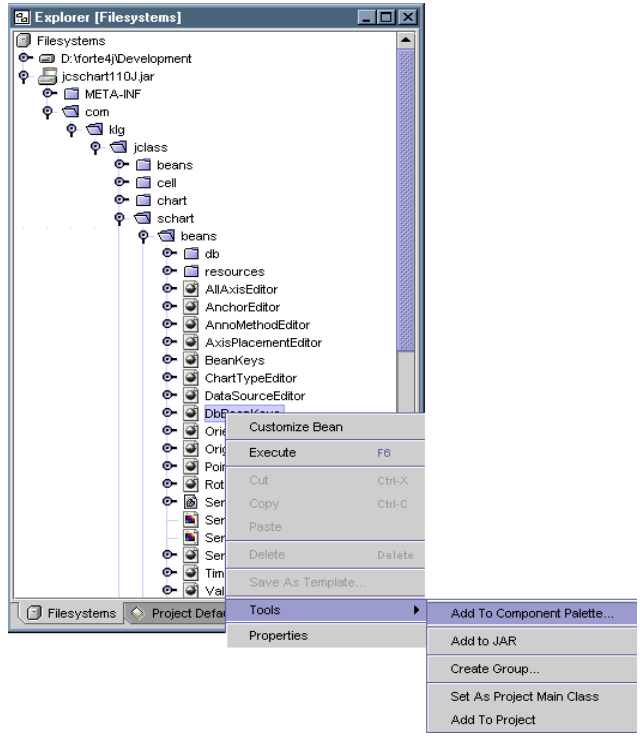
   or

   - Go to **File/Mount Filesystem** and select the **Mount JAR File** (or **Add JAR File**, depending on your edition of Forte) checkbox from the **Mount Filesystem** window. Use the **Browse** button to navigate to the location of the JClass JAR files, and select the one(s) you would like to install.



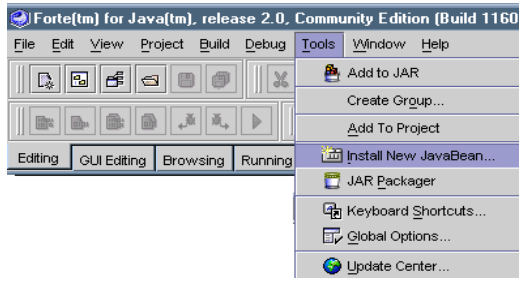
2. After mounting the JAR, select the Bean(s) you would like to add. This can be done in one of two ways:

   ■ In Forte's **Explorer** window, drill down to the JClass Beans directory and right click on the Bean(s) you would like to install. Select **Tools/Add to Component Palette...** from the menu.



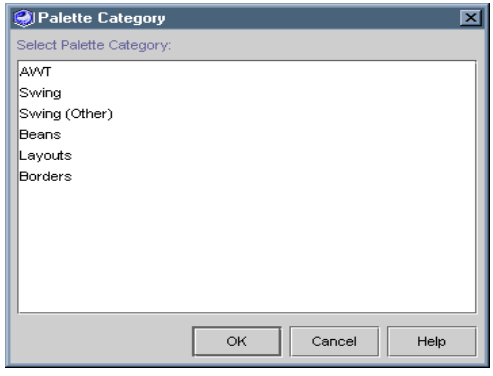
   or

   ■ Go to **Tools/Install New Java Bean...**



   Navigate to the directory that contains the JAR files and select the Bean(s) you would like to install from the list. Click **OK**.

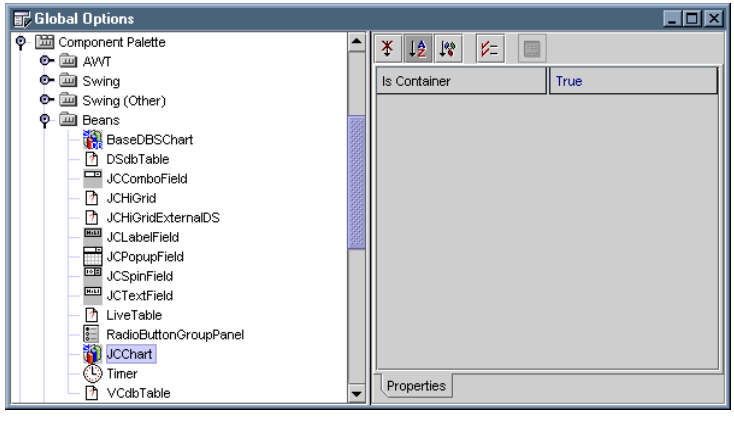
3. Assign the integrated Beans to a category from the **Palette Category** window, for example, Beans.



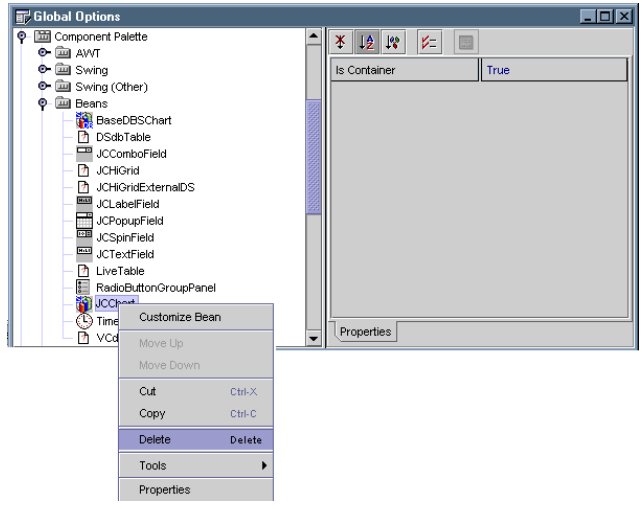
4. Your Beans will be located on the right side of the toolbar, under the tab that corresponds to the category you chose.

Removing JClass JavaBeans from Sun's Forte for Java

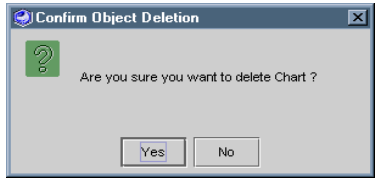
1. To remove JClass Beans from Sun's Forte for Java, go to the **Tools** menu and select **Global Options** (or **Options**, depending on your edition of Forte).
2. In the **Global Options** (or **Options**) window, view the Component Palette and open the palette category where your Bean is located (eg, JCChart and JCHiGrid).



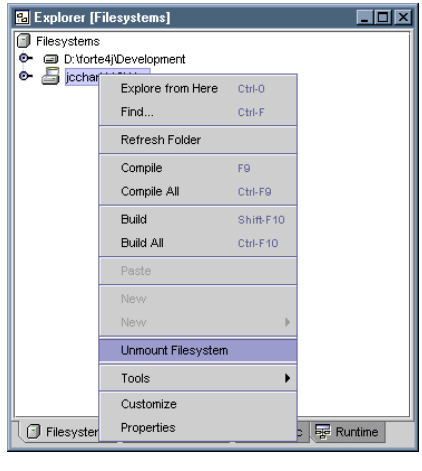
3. Right click on the Bean you would like to remove and select **Delete** from the drop-down menu. (You can also select the Bean and hit the Delete key.)



4. When the **Confirm Object Deletion** window appears, click **Yes**.



5. Next, you must remove the JClass JAR from the filesystem. To do so, right click on the filesystem you want to remove (the JClass filesystem) in Forte's **Explorer** window and select **Unmount Filesystem** from the menu. (You can also unmount the filesystem by going to the **File** menu and selecting **Unmount Filesystem**.



Technical Support

Pre-Sales Support ■ *Technical Support*
Product Feedback and Announcements

Pre-Sales Support

Customers that are evaluating JClass DesktopViews products can receive pre-sales support. We are happy to provide consultation on the features and capabilities of any JClass product. Note, however, that priority will be given to Gold Support customers.

Technical Support

Many of the initial questions you may have are concern basic installation or configuration issues. Consult this product's *readme* file and the *Getting Started Guide* (available in HTML and PDF formats) for help with these types of problems.

Sitraka's **Gold Support with Subscription** plan is included with your purchase and entitles registered users with a valid JClass software license to the following support:

- Product documentation, API reference, and demos and examples, included with the product and/or downloadable from our Web site, and/or available online.
- All product upgrade releases; download from our Web site.
- FAQ Documents on our Web site.
- JClass Knowledge Base, a searchable collection of information including program samples and problem/resolution documents.
- SupportWatch, a convenient way to log and track support requests over the Web.
- Direct technical support for one full year.
- JClass Forum Newsgroup, where you can communicate with other developers using JClass products around the world.

For additional information and pricing for JClass **Gold Support with Subscription**, please visit our online store or your JClass reseller. You can also email sales@sitraka.com.

To Contact JClass Support

Any request for support *must* include your JClass DesktopViews product serial number. Supplying the following information will help us serve you better:

- Your name, email address, telephone number, company name, and country
- The product name, version, and serial number
- The JDK (and IDE, if applicable) that you are using
- The type and version of the operating system you are using
- Your development environment and its version
- A full description of the problem, including any error messages and the steps required to duplicate it

You may also use our online email form to submit the above, available at *http://www.sitraka.com/software/support/jclass/tsjclasssupport.html*

JClass Direct Technical Support	
SupportWatch (Web-based support tool)	*https://supportwatch.sitraka.com* (to help protect the confidentiality of your information, SupportWatch is provided over a secure Internet connection)
JClass Support Email	*jclass_support@sitraka.com*
Telephone	800-663-4723 (toll free in North America) or 416-594-1026 Available Monday – Friday, 9:00 a.m. to 8:00 p.m. EST
Fax	416-594-1919
European Customers Contact Information	Email: *eurosupport@sitraka.com* Telephone: +31 (0)20 510-6700 Fax: +31 (0)20 470-0326 Available Monday – Friday, 9:00 a.m. to 5:00 p.m. CET
Other Support Resources	
Using JClass in IDEs	*http://www.sitraka.com/software/jclass/jclassides.html*
JClass FAQs	*http://www.sitraka.com/software/support/jclass/tsjclassfaq.html*
JClass Technical Support (links to Knowledge Base)	*http://www.sitraka.com/software/support/jclass/tsjclasssupport.html*
JClass Forum Newsgroup	*http://newsweb.sitraka.com/cgi-bin/dnewsweb/*

Product Feedback and Announcements

We are interested in hearing about how you use JClass DesktopViews, any problems you encounter, or any additional features you would find helpful. The majority of enhancements to JClass products are the result of customer requests.

Please send your comments to:

Sitraka
260 King Street East
Toronto, Ontario, M5A 4L5 Canada

Phone: 416-594-1026
Fax: 416-594-1919
Email: *jclass_suggestionbox@sitraka.com*
Internet: *http://newsweb.sitraka.com/cgi-bin/dnewsweb/*

While we appreciate your feedback, we cannot guarantee a response.

Occasionally, we send JClass-related product announcements to our customers using an email list. To add yourself to this mailing list, send email with the word "`subscribe`" in the body of the message to *javanews-request@sitraka.com*. Visit the Sitraka Web site at *http://www.sitraka.com* for more details.